2.0 Conceptual Model Specification

The Conceptual Model Specification (CMS) has been extracted from the *Post-Development Design Document* (PD^3) [A.2-3] for ALARM and is intended to be a substitute for the Software Design Document (SDD) called for in DOD-STD-2167A. ALARM is a mature model, written before this standard was in place; therefore it did not have an SDD before SMART V&V efforts produced a post-development substitute for it. The CMS delineates both high-level and detailed design descriptions and requirement specifications for ALARM. Thus, it serves as the basis for model verification. It also enhances the user's understanding of the model. The progression of the model descriptions from the very general (operational concept) to the very specific (detailed design) should facilitate comprehension of ALARM.

The CMS is being developed in stages, and is not yet complete. The current stage includes operational concept and high-level design information as well as detailed design for several functional elements (FEs). Detailed design information for other FEs will be incorporated as changes to this document.

Operational concept, design, and requirements specifications are being determined from several key sources. Top-level mission and system design information is consolidated from all available model developer documentation such as User's, Analyst's, and Programmer's manuals. The primary sources for detailed design information are the Verification Source Reports (VSRs) produced by the model developer as inputs to the PD^3. The VSRs provide detailed information about code and algorithm design. Additional information may be obtained from Computer-Aided Software Engineering (CASE) tool analysis and direct code examination.

2.0.1 Operational Concept

The Advanced Low-Altitude Radar Model (ALARM) is a digital computer simulation designed to determine the detectability of a single target by a single radar. ALARM was originally developed for the purpose of examining the benefits of low-altitude flight for minimizing detection of air vehicles.

The primary missions of the model are to provide an evaluation of the capability of a ground-based radar system to detect a low-altitude air vehicle, and to aid analysts in the study of detectability phenomena. The secondary missions of ALARM are to generate output data for higher-order models and to produce detection contours.

ALARM can simulate three types of radar systems: pulsed/MTI, pulse doppler, and continuous wave (CW).

ALARM models the effects of atmospheric attenuation, terrain masking, clutter, multipath, electromagnetic propagation, noise, coherent self-screening jammers, stand-off noise jammers, pulse eclipsing/blanking, and MTI range and azimuth gating. The range of radar frequencies ALARM can model is from 10 to 35000 MHz.

The user can choose one of two modes for the target, and one of two options for terrain processing. The two available target modes are flight path and contour. In flight path mode, a table of flight data is input to the model. Each flight path point specifies values for altitude, heading, velocity, bank, and pitch. Detection is checked at each flight path point during execution of ALARM. The processed output from flight models such as BLUEMAX can be input to ALARM under flight path mode. In contour mode, ALARM simulates multiple straight and level flight paths implemented by an array of equally-spaced up/down range and cross range position points. The target speed, heading, and attitude are constant for all target positions. For both modes, a table of aircraft RCS values can be input. In either mode, ALARM determines a signal-to-interference ratio (S/I) for each target position and outputs a binary file containing signal-related data. In flight path mode, a print-format file, containing data about each point, is generated.

ALARM may be run in either a site-specific (terrain) or round smooth earth mode. In site-specific mode, the location of the radar is used along with Defense Mapping Agency (DMA) Digitized Terrain Elevation Data (DTED) to simulate masking conditions at actual geographic locations around the globe. In either mode, ALARM determines the clutter reflection from clutter patches using optical line-of-sight calculations. This method also utilizes a set of bivariate clutter reflectivity density distributions published by MIT Lincoln Laboratories.

Limitations to Model Use

ALARM is restricted to a one-vs-one engagement of a fixed-location aircraft and an airborne target.

ALARM is a target position-stepped model that considers instantaneous target conditions. Since the model is not time-stepped, radar and target dynamic functions are not explicitly modeled. For example, perfect range, angle, and velocity tracking are assumed.

While the primary purpose of the model is the detection of low-altitude targets, higher altitude targets can also be used, with the limiting factor that only one radar beam can be modeled. For multiple beam radars, modeling only the lowest beam is in most cases adequate for evaluating the radar's capability against low-flying targets. For higher altitude targets, modeling the low beam should provide the maximum detection capability of the radar; however, gaps may appear closer to the radar since this area is normally covered by higher-angle beams.

ALARM provides only a limited capability to simulate CW type radar systems; CW is modeled as a special case of the pulse doppler radar. In addition, some radar features found in specific radar types (e.g., STC and CFAR) are neither explicitly nor implicitly modeled.

It is assumed that the analysts utilizing ALARM are generally familiar with radar and aerodynamic theory so that the user-supplied input blocks are reasonable. The input consists of engineering-level data such as transmit power, pulse width, pulse repetition frequency, two-dimensional antenna patterns and RCS tables, and data needed to simulate MTI and pulse doppler processing. ALARM checks that the input data blocks are complete and that values are within specified limits; however, the data are not evaluated for engineering-level validity.

2.0.2 Top-Level Software Design

In terms of functional decomposition terminology, ALARM has only one major component, the RF Sensor. The RF Sensor implemented in ALARM includes these functional areas: target characteristics, propagation, transmitter, receiver, antenna, and signal processing.

ALARM is an integration period model; signal processing is accounted for by deterministic mathematical formulation or by recursive algorithm. ALARM distinguishes among the radar types it models (CW, pulsed, and pulse doppler) primarily on the basis of target and clutter signal filtering. A Chebychev filter response function is used to model the response of CW and pulse doppler radars to target and clutter signals, and an MTI filter function models the response of pulsed radars to these signals. The parameters matching the appropriate filter to the type of radar being modeled are part of the input.

A variety of different scenarios can be simulated in the model by changing various input parameters, such as flight path, RCS tables, and jamming sources. ALARM incorporates the Swerling-Barton technique to model detectability of different classes of fluctuating targets.

ALARM models the following propagation phenomena: scattering of energy from the earth's surface or from clutter objects; atmospheric attenuation and refraction of radar signals; specular reflection and diffraction of radar signals; random noise.

Model Conventions

There are several general assumptions and limitations used in the construction of ALARM 3.0. These are addressed below.

a. Time. Time is measured in seconds, or decimal fractions of seconds.

b. Distance. Distance is measured in meters.

c. Real and Complex Number Precision. Real and complex number default precision in the code is double precision.

d. Platform Dependency. There could be some small differences in the execution of the ALARM code compiled on different computer or compiler types. The only known problem of this sort is in the compiler/linker library random number routine: there is no standard random number generator in FORTRAN 77. The only way to avoid this kind of difference would be to build an ALARM-specific random number generator, distributed as part of the baseline code.

Logic Flow Through Major Components

The flow of control through ALARM 3.0 is depicted in figure 2.0-1. There are four major areas of processing that occur within the model: initialization, contour mode processing, flight path mode processing, and generation of output.

The first phase of model execution is initialization of the model run; this includes input, error-checking, and data initialization. The input data created by the user for the current execution are checked for errors and are then used to initialize various common blocks accessed by ALARM. The input parameters are discussed in the next section, Data Flow.

Following successful initialization of the run, ALARM execution proceeds for the type of radar and flight mode selected in the input. As the Flow Diagram shows, the logic for the two different flight modes is very similar:

1. parameters for the current target position, relative to the radar, are calculated;

2. interference sources, including system noise, deceptive jamming, and clutter, are processed;

3. the signal-to-interference ratio (S/I) and probability of detection (P_d) are calculated; and

4. a decision is made to end processing (if all flight path points have been processed) or to continue (if any remain).

The details of the specialized processing that occurs for the type of radar system selected by the user are contained within the process boxes in the diagram. In the "Initialize Model Run" box, the program calls the appropriate subroutine to handle either a pulsed or pulse doppler radar. The calculations that are performed for the interference sources, in the box labeled "Process Interference Sources," are radar-specific. In the case of pulsed radars, MTI processing is conducted. For pulse doppler radars, calculations are made for each of the PRFs defined in the input, and the estimated P_d is then based on the maximum S/I computed.

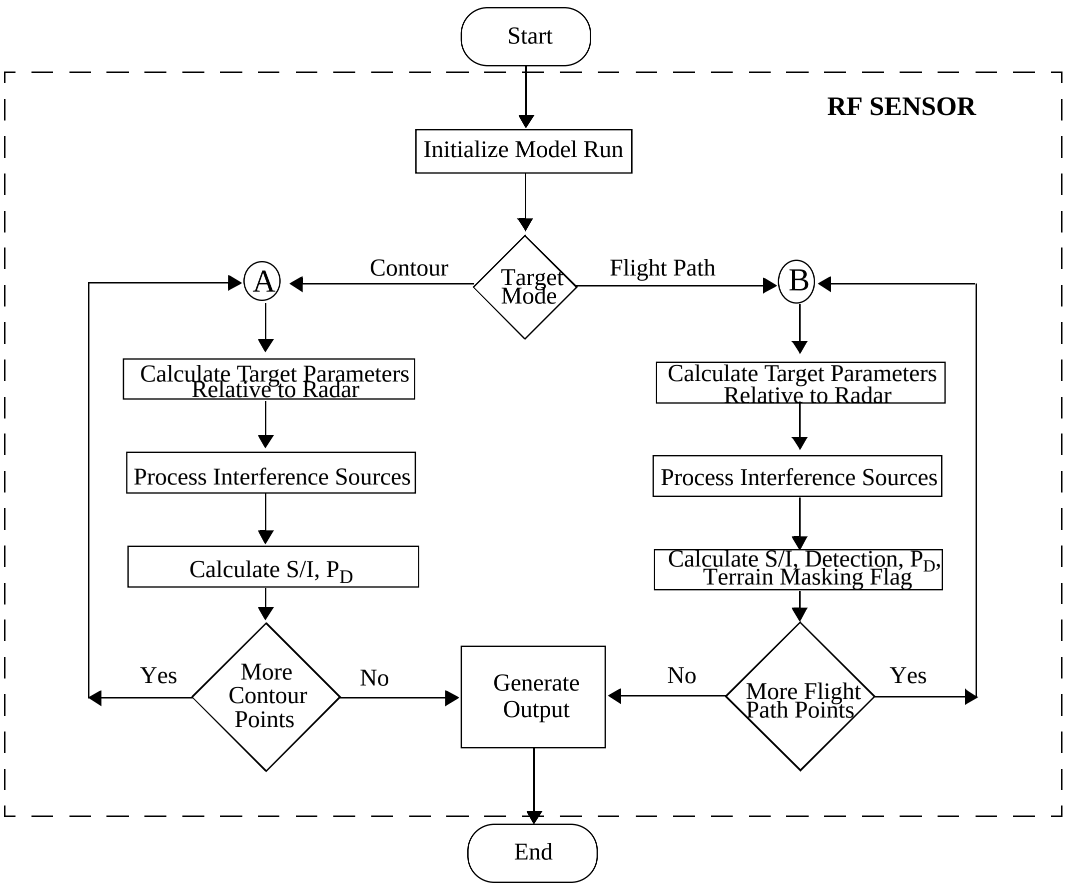

Figure 2.0-1 Logic Flow Diagram

After all processing has occurred, ALARM execution concludes with generation of the output. Two possible types of output are available, depending upon the flight mode selected. For contour mode, data for generating a contour plot is produced. For flight path mode, a report showing S/I, detection flag, and masking flag for each flight path point is produced.

Data Flow Through Major Components

The input to ALARM consists of 13 independent data blocks, specified in the main input file as actual values or the pathname of the file containing the values. Each data block is briefly described below:

DATAGANR: Radar receiver antenna pattern.

DATAGANT: Radar transmitter antenna pattern.

DATAJAMR: (Optional) Types of jammers and their power values.

DATAPLOT: Target information for running the model in contour mode; either this or the DATATARG data block, but not both, must be defined.

DATATARG: Target information for running the model in flight path mode; either this or the DATAPLOT data block, but not both, must be defined.

DATARCST: Table of RCS data for the target.

DATARADR: Type of radar mode and engineering-level data for the selected radar type, including such values as transmit power, pulse width, and pulse repetition frequency.

DATAREFL: MIT Lincoln Labs clutter reflectivity data.

DATAROTO: (Optional) Helicopter rotor spectral data.

DATASEKE: Radar propagation data.

DATASITE: Location of radar site; this block is required, whether running the model with or without terrain.

DATABASE: Terrain data base information. (Required when running the model with terrain.)

DATATITL: ALARM run title.

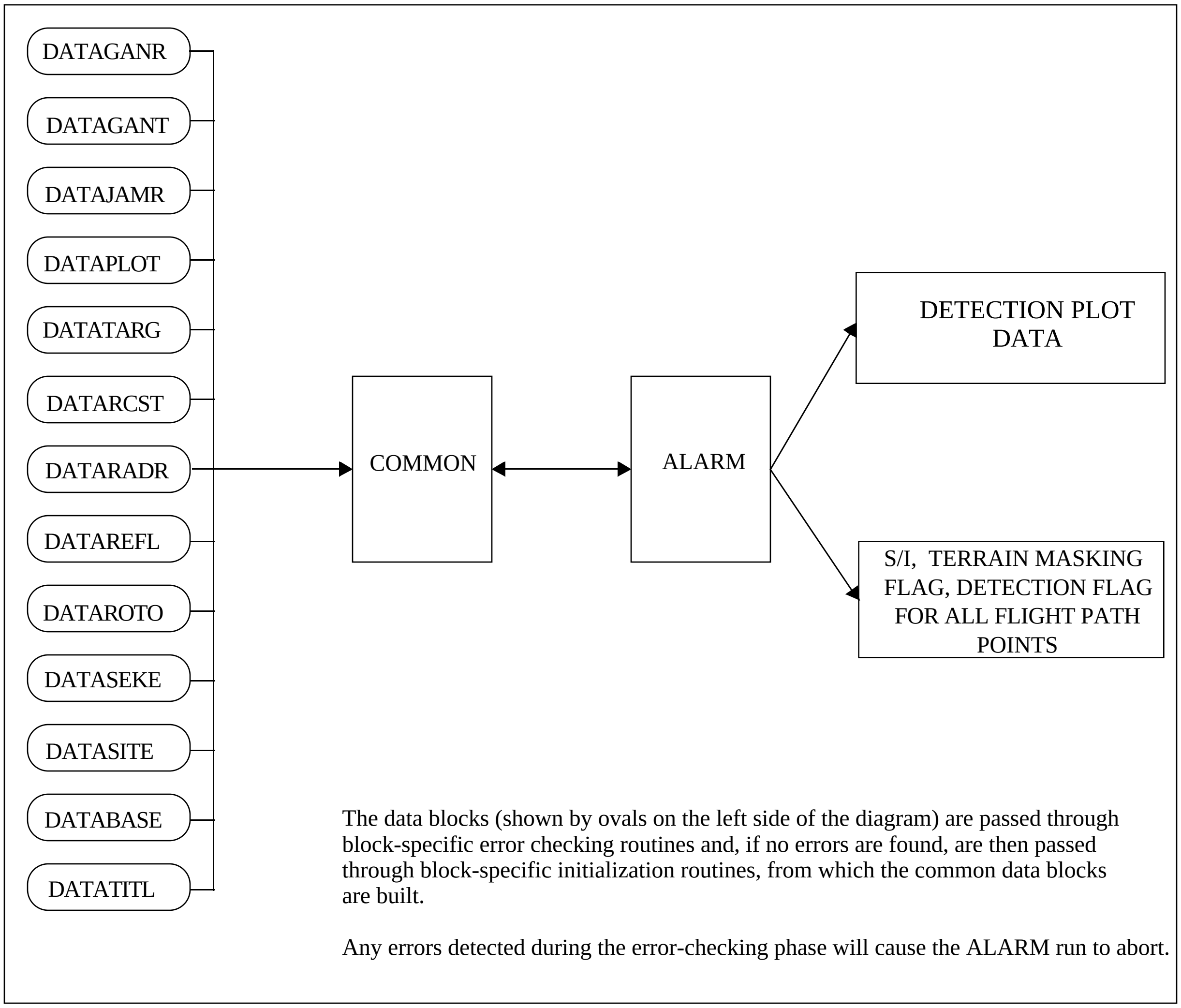

Figure 2.0-2 Data Flow Diagram

As shown in figure 2.0-2, the movement of data into and out of ALARM's single major component is quite simple. The input data blocks are read, checked for completeness and range limits, and then used to initialize common blocks accessed at a global level by the subroutines within the program. Output from the model consists of either data for a contour plot depicting a detection pattern or a report of significant values calculated during execution.

Source Code Hierarchy

A source-code hierarchy chart for ALARM is shown in figure 2.0-3. The main program, named ALARM, calls initialization routines, followed by the processing subroutine, RADCAL, and output generation subroutine, OUTPUT. RADCAL calls either PULSED or PULDOP, depending upon the radar type selected by the user. The RF sensor functions are implemented in RADCAL and the routines it accesses; each is briefly discussed below.

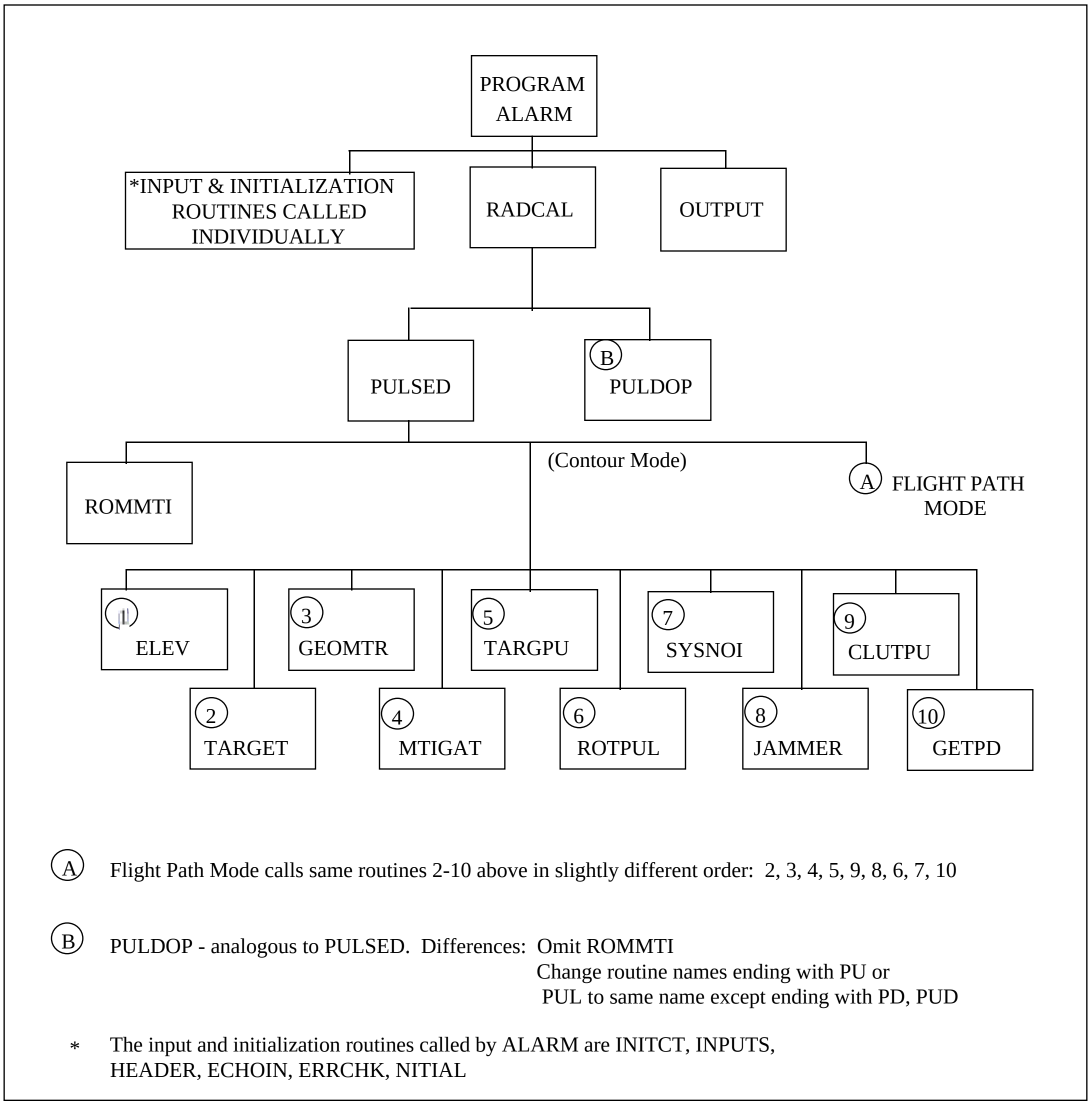

Figure 2.0-3 Source Code Hierarchy

The initialization routines are:

ALARM: Program ALARM is the main routine for the program.

INITCT: Initializes global constants.

INPUTS: Controls data input of ALARM. The primary input file is read sequentially, and input of the data blocks encountered is performed by the appropriate input subroutines. Unknown data block names encountered are listed in the printed output file.

HEADER: Places a banner page near the beginning of the printed output. This page contains the data and time of the ALARM run and the title of the run.

ECHOIN: Calls the print routines for each of the data blocks to print the inputs contained in each data block to the output file.

ERRCHK: Performs printing to the output file of errors discovered in the input data blocks. ERRCHK calls a separate subroutine for each input data block.

NITIAL: Calls individual subroutines to initialize each input data block. Initialization includes converting between measurement systems and determining new variables from the user-supplied data.

The processing routines are:

RADCAL: Calls the appropriate subroutine for radar processing, either PULDOP for pulse doppler radars or PULSED for pulsed radars.

PULSED: The program execution subroutine for a pulsed radar. It performs the final determinations of S/I and detection.

PULDOP: The program execution subroutine for a pulse doppler radar. It performs final determination of S/I and detection.

ROMMTI: Integrates the clutter signal within the limits passed as parameters.

ELEV: Determines the terrain elevation in meters above mean sea level (MSL).

TARGET: Computes a composite transformation matrix of the target based on a simple transformation and an orientation matrix. This matrix is used to determine the viewing aspect angles of the target.

GEOMTR: Slews the antenna to the target within the user-specified limits, then calculates the off-boresight and viewing aspect angles.

MTIGAT: Determines if the target is within one of the specified azimuth/range MTI gates.

TARGPU: Determines the target signal return after modification by MTI attenuation, pattern propagation factor, atmospheric attenuation, and eclipsing loss. Also calculates the onboard jammer signal's pattern propagation factor.

ROTPUL: Determines, for a pulsed MTI-type radar, the contribution to the target radar cross section due to rotating rotor blades.

SYSNOI: Computes the total radar system noise temperature including the noise contributions from the antenna and the receiver.

JAMMER: Computes the jamming signal power of an onboard deception jammer, an onboard self-screening jammer, a stand-off noise jammer, or combinations of the three types of jammers.

CLUTPU: The clutter determination subroutine for a pulsed radar.

TARGPD: Calculates the target signal power at each PRF, atmospheric attenuation and pattern propagation factors, and the onboard jammer signal's pattern propagation factor.

ROTPUD: Determines, for a pulse doppler type radar, the rotor blade signal.

CLUTPD: The clutter determination subroutine for a pulse doppler radar.

2.0.3 Top-Level Requirements

The Advanced Low Altitude Radar Model (ALARM) shall be a digital computer simulation designed to determine the detectability of a single target by a single radar.

ALARM shall simulate these three types of radar systems:

1. pulsed/MTI,
2. pulse doppler, and
3. continuous wave (CW).

The pulsed radar type shall include MTI processing, and the pulse doppler radar type shall handle multiple pulse repetition frequencies (PRFs).

ALARM shall distinguish among the radar types it models on the basis of target and clutter signal filtering. The parameters matching the appropriate filter to the type of radar being modeled shall be part of the input.

The range of radar frequencies ALARM will model shall be from 10 to 35000 MHz.

ALARM shall have only one major component, the RF Sensor. The RF Sensor implemented in ALARM shall include these functional areas: target characteristics, propagation, transmitter, receiver, antenna, and signal processing.

Target Characteristics

ALARM shall provide a flight path mode, chosen by the user, in which to execute the model. In flight path mode, a table of flight data shall be input to the model. Each flight path point shall specify values for altitude, heading, velocity, bank, and pitch. Detection shall be checked at each flight path point during execution of ALARM. ALARM shall allow the processed output from flight models such as BLUEMAX to be accepted as input under flight path mode.

ALARM shall provide a contour mode, chosen by the user, in which to execute the model. In contour mode, ALARM shall generate multiple straight and level flight paths, based on single input values for aircraft velocity and altitude.

For both flight path and contour mode, a table of aircraft RCS values shall be input.

ALARM shall model coherent self-screening jammers and stand-off noise jammers.

Propagation

ALARM shall be run in either a site-specific or round smooth earth mode.

In site-specific mode, the location of the radar shall be used along with Defense Mapping Agency (DMA) Digitized Terrain Elevation Data (DTED) to simulate masking conditions at actual geographic locations around the globe. In site-specific mode, ALARM shall determine the clutter reflection from clutter patch pixels using optical line-of-sight calculations. This method will also utilize a set of bivariate clutter reflectivity density distributions published by MIT Lincoln Laboratories.

ALARM shall specifically model these propagation phenomena: scattering of energy from the earth's surface or from clutter objects; atmospheric attenuation and refraction of radar signals; specular reflection (multipath) and diffraction of radar signals; and random noise.

Transmitter, Receiver, and Antenna

ALARM shall model the transmitter of the radar systems it simulates.

ALARM shall model the receiver of the radar systems it simulates, including thermal noise, perfect automatic gain control (AGC), detector type, and pulse blanking/eclipsing.

ALARM shall model the antennae of the radar systems it simulates, and allow the user to define antenna patterns as tables of gain values for both the transmitter and receiver.

Signal Processing

ALARM shall be an integration period model (i.e. signal processing shall be accounted for by deterministic mathematical formulation or by recursive algorithm).

The calculations and decisions that are made within the ALARM program shall be tailored for the type of radar, flight path mode, and terrain option selected by the user.

ALARM shall model MTI and doppler filters.

ALARM shall model pulse compression.

Output

ALARM shall generate two possible types of output, depending upon the flight mode selected. For contour mode, data for a detection contour plot will be generated. For flight path mode, a report showing S/I, detection flag, and masking flag for each flight path point shall be produced.

Implementation Requirements

ALARM shall be written in FORTRAN according to FORTRAN-77 standards.

ALARM shall be hosted on these computer systems: VAX/VMS, UNIX, IBM-compatible personal computers.

For VAX/VMS systems, ALARM shall require 3 MB of disk storage space to host the basic ALARM installation package. This package shall contain the executable source and source code, sample input and output files, and all command files needed to compile, link, and execute the model.

Each one-degree by one-degree DMA DTED file shall require additional disk storage ranging from approximately 480 KB to 2.88 MB, depending on the latitude band containing the file.